Physics-Based Modeling as an Alternative Approach to Geometrical Constrain-Modeling for the Design of Elastically-Deformable Material Systems

Moritz Fleischmann[1], Achim Menges[2]
Institute for Computational Design (ICD), Stuttgart University.
http://icd.uni-stuttgart.de
[1]moritz.fleischmann@icd.uni-stuttgart.de, [2]achim.menges@icd.uni-stuttgart.de

Abstract. *Physics-Based Modelling can be considered as an alternative approach to geometrical constrain-based modelling for form-active material systems such as gridshells. Here we explain a vector-based method that works in R2 and R3 to determine momentum forces at the node level, which can easily be implemented into (existing) particle systems and - together with the simulation of tension and compression forces - can be used to model the behavior of such material systems.*
Keywords. *Computational Design; Physics-Based Modelling; Springs; Bending; Material Behaviour.*

INTRODUCTION

Through the parametric extension of CAD software, computer-aided modeling has gained a lot of attention within the architectural design community in recent years. This interest is understandable from the perspective of a designer who is seeking a formal exploration of geometric shapes. Unfortunately any strictly geometric method renders itself to be unsuitable for the design of force-active material systems. Here, form is inseparably connected to the material characteristics of the system and the forces applied to it.

In Architecture, well-known examples of built material systems include tension-active structures such as cablenets & membranes (Otto and Schleyer, 1962). Similarly bending-active structures (Knippers et. al., 2010) such as gridshells (Walser 2011; Otto et. al., 1964) require a different approach in terms of computer-aided design: In order to investigate their forms, the designer needs to incorporate actual physical behavior into his design model (Figure 1).

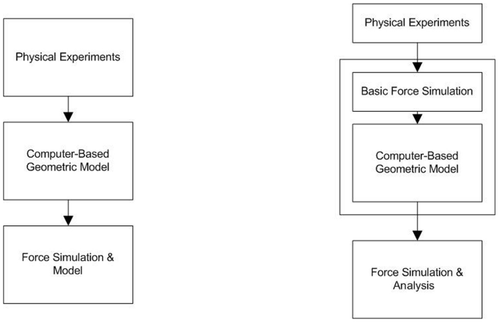

Physical Experiments
Computer-Based Geometric Model
Force Simulation & Model
Physical Experiments
Basic Force Simulation
Computer-Based Geometric Model
Force Simulation & Analysis

Figure 1
Left: Established Design framework for the design of force-active material systems. Right: Proposed extended design framework.

Historically, the development of physical form finding techniques arose out of the inability to capture the behavior of such material systems in drawings, plans and sections (Gaß, 1990). Today we are facing a similar dilemma: Due to their inherent geometric nature, the computer-aided design tools we as designers are so familiar with seem to be inappropriate for the design of form-active structures. Even though sophisticated methods for the simulation of structural behavior exist (Fröhlich, 1995), they are

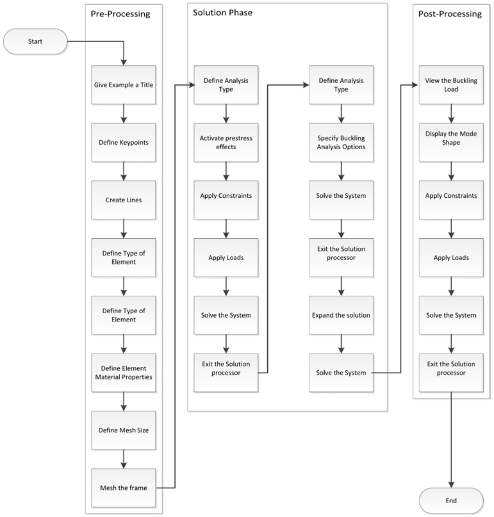


Figure 2
Flowchart depicting the process of setting up a FEM analysis for buckling simulation of linear spline beam elements in ANSYS - too many steps for quick assessments with knowledge of specific material parameters.

seldom implemented into the design modelling Environments.

The inaccessibility of engineering-oriented simulation software renders itself often unsuitable for design variation as it requires a relatively large set of input parameters to run and specific knowledge about materials, structural systems and loading in order to operate (Figure 2).

The linearity of current "modeling and simulation" strategies, where geometric form is determined prior to a subsequent steps of analysis is unsuitable from a designer's (with an interest in force-active systems) perspective.

In the following paragraphs, we look at an alternative modeling approach to the previously mentioned geometric constraint modeling, namely physics-based modeling. After explaining the functionality and some of the noteworthy architectural implementations of this technique, we highlight benefits and limitations. We look at recent (software-) developments and furthermore we aim to explain the repercussions this alternative technique has on the interrelation between the designer and the model.

On the example of the design of a bending-active structure, we compare the physics-based modeling approach to a parametric modeling approach based on physical form finding experiments.

EXAMPLES OF PHYSICS-BASED MODELING IN ARCHITECTURAL DESIGN

Physics-Based Modeling (Eberly, 2004) is a well-established field within the scientific research of computer science. In general Physics-Based Modeling describes the construction of dynamic models of animated objects and computing their motions via physical simulation. Physics-based modeling implies that object motions are governed by the laws of physics, which leads to physically realistic animation. One of the most noteworthy early contributions in the field of Physics-Based modeling with a strong connection to tension-active systems was Baraff and Witkin's presentation at the 1998 Siggraph conference (Baraff and Witkin, 1998). Many principles described in this paper were adopted for the development of numerous Physics-Engines and led to the development of various computational integration methods which form the core of any physics engine (Eberly, 2004; Gibson and Mirtich, 1997).

What are particle systems?

Particle systems, a term first coined by William T. Reeves in 1983 (*as he worked to create the "Genesis" effect at the end of the movie, Star Trek II: The Wrath of Khan*), are a simple implementation of a physics-based model. They have been used as a device for design exploration within many architectural projects, but not often as a device to mimic real physical behaviors. A particle system can be considered a collection of independent objects, often represented by a simple shape or dot. It can be used to model many irregular types of natural phenomena, such as explosions, fire, smoke, sparks, waterfalls, clouds, fog, petals, grass and bubbles.

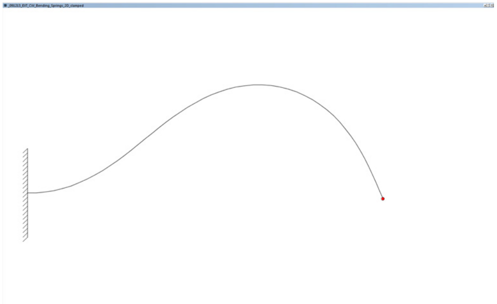


Figure 3
Reference: 2D-Simulation of buckling in Processing with Verlet Integration (Source: C. Williams).

Implementations of particle systems in architectural design environments

When John Ochsendorf (together with Axel Kilian, Barb Cutler, Erik Demaine and Marty Demaine) conducted the Workshop "Exploring Gaudi's World" [1] in 2004 it was noteworthy for the fact that it was the first time that a collaborative effort was made to utilize a particle system to explore architectural systems such as catenary networks, that previously depended on building elaborative physical models (Gaß, 1990). The Particle System utilized was very simple, yet led to remarkable results. (Kilian and Ochsendorf, 2005)

While the Particle System written for this class was never published, a lot of other Particle Systems and physics engines were developed for Processing. The developers of these engines did not build these engines with an architectural user group in mind, yet their functionality is not limited to mere simulation of visual effects, as the MIT workshop showcased already.

Within a particle system there are various types of forces, a very common one being the spring-force. Usually these forces act between a set of 2 particles either pushing or pulling them together. In 2009 Chris Williams has published a Verlet integration based Processing sketch [2] that is able to simulate bending behavior by implementing method to calculate the internal force at each node of a buckling linear beam element (Adriaenssens, 2005) (Figure 3).

The same forces and integration method (as well as 3 others) have been incorporated by Daniel Piker [3] into his freely available Plugin "Kangaroo" [4] for Rhino Grasshopper [5]. Even though none of the benefits of the object-oriented framework a platform such as Processing provides, the tool has been very positively received by the GH-User community and is under constant development.

In parallel there has been an increasing interest in the development of novel simulation techniques such as the Thrust Network Analysis (Block and Ochsendorf, 2007). At the same time the relationship of Cullman's method of graphic statics to parametric design are being explored (Lachauer et. al., 2011).

While the previous research focusses on abstracting structural dependencies into vector-based analytical models, there have been also attempts to implement real-time physics as part of a generative design process. (Ramtin et. al., 2010)

What all of the simulation tools (except the analytical ones based on graphic statics) have in common, is that they form another layer between the designer and the model. This additional layer is on the one hand necessary to simulate the physical behavior of the system. On the other hand, it makes it difficult to produce meaningful results for the inexperienced user, because the designer's judgment is not only challenged by decisions on form, but also structural behavior and materiality. The question, how a certain parameter such as a springs stiffness relates to a "real" physical materials stiffness cannot be answered without an in-depth knowledge of the mechanisms of the specific physics engine as well as the general limitations that apply due to the high degree of idealization for the sake of speed and efficiency commonly implemented in most physics-engines.

We have expanded Chris Williams' 2-dimensional Processing sketch into a 3D simulation environment in Processing and conducted a set of comparisons between the out-of plane buckling behavior of thin physical rods (mechanical splines) and the simulated model.

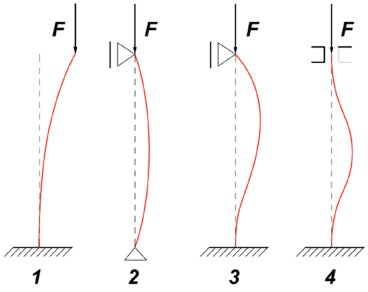


Figure 4 (left)
Euler cases 1-4: Simulation of generic formal behaviour based on node constraints.

SIMULATION OF GENERIC FORMAL BEHAVIOUR VS. SIMULATION OF SPECIFIC MATERIAL BEHAVIOUR

"Simulation" in the context of architecture is often referred to the precise simulation of physical properties by means of CAE software. "Structural simulation" aims to calculate, visualize and communicate the distribution of stresses within a given geometry under external loads. "Formfinding" is a term that was developed as part of the investigations of lightweight structures and focusses on similar topics with the difference being that geometry is allowed to undergo large deformations in the process of simulation. These large deformations have led to the development of numerous simulation algorithms.

What the previously outlined CAE-oriented structural simulation models have in common is that they depend on a (virtual) material input (e.g. wood), which is then linked to further software-immanent information such as young's modulus and non-linear behaviour under stress. Combined with geometric information such as cross-sectional data, this input is used to simulate material Behaviour (with regards to deformation) under external loads.

An alternative to the simulation of form based on the (specific) material behaviour is the simulation of (generic) FORMAL behaviour. This method is less precise in the sense that it does not account for specific material properties such as deformation, deflection, failure, stresses etc., but it is never the less as accurate, because all material things share certain formal behaviour when exposed to external loads (Figure 4).

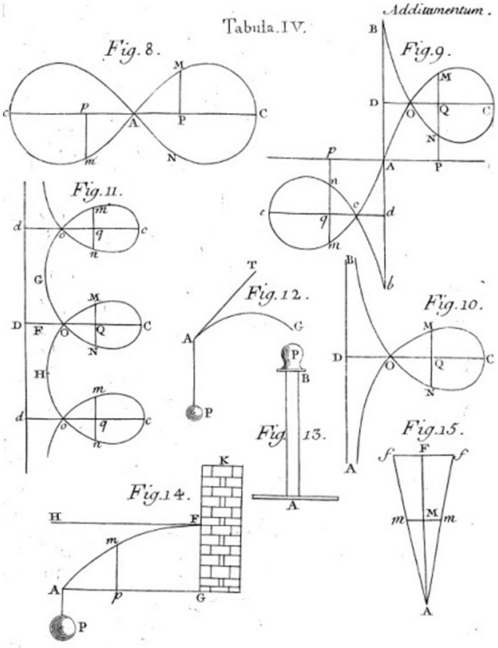


Figure 5 (right)
Euler's elastica figures, Tabula IV.

Example: The catenary is a generic formal description of a (physical) chain acting under gravity and tension. Here, material is not essential to simulate formal behaviour, as the mathematical description of the catenary does not need any material input to compute. It has been shown, that Particle-Systems can be used to approximate the behaviour of catenaries and networks thereof.

This way of "form-finding" presents an interesting approach to the design of form-active systems from a designer's point of view: As form and geometric behaviour can be simulated using positions rather than information about material and geometry. Therefore, the simulation can be run with very few input parameters while maintaining accuracy. This makes them appealing from a designer's perspective who is interested in formal explorations without knowing every detail of the final design yet. Opposed to highly specific structural simulations, this lack of precise inputs poses a challenge to the designer, because the few simulation parameters (e.g. a spring's stiffness) do not link as directly to specific physical material properties - here the designers expertize is required or the modelling environment needs to be refined as the design progresses.

PHYSICS-BASED SIMULATION OF FORM

What the catenary is to tensioned systems and compression-active systems, is the elastica (Levien, 2009) to bending-active systems. At least it has the potential. It has not been investigated to similar extends yet within the architectural design community and there are less examples of built bending-active systems in architecture compared to cablenets and compression-active vaults.

A characteristic form of the elastica is its "drop-like" shape, when the two ends of a elastica with l>0 meet (Figure 5).This characteristic shape prohibits the use of explicit mathematical descriptions (where every value x has 1 associated y value ONLY).

Here, we would like to "skip" the step of the mathematical description of the elastica (Levien, 2009) and focus on the physics-based model - an expansion of the particle-system already proven to be successful for the design of tension-active systems, expanded by an additional vector that at accounts for bending force. In such a system, the curve is discretized into line segments, which act as springs. If we stopped here, we would have a virtual chain model. What needs to be added in order to approximate an elastica in R2 is an additional force vector at each node. It has been shown how to calculate this momentum vector based on the angle at each node (Adriaenssens, 2001).

Another very effective way to determine this vector in R2 has been implemented by Chris Williams in Processing. The length of the vector at each node, that forces the curve to flatten, is twice the distance between the node's current position to the midpoint of both of its neighbors (Figure 6 and 8).

Because three nodes are always coplanar (lie in the same plane), it is acually possible to expand this model from R2 to R3. Experiments show that the overall form of a spatial elasica curve can now be approximated. Yet the twist of the curve is not simulated as no force is transferred from one line segment to the next. While it would be relatively easy, in programming terms, to achieve this - and there have been numerous publications in the Computer Graphics Industry on the topic of modelling elastic deformations of linear elements recently (Gibson and Mirtich, 1997) - we would like to promote an alternative approach:

Figure 6 (left)
Vector-based approach to determine momentum Vector (M) at any node (B) in a finite element elastica model.

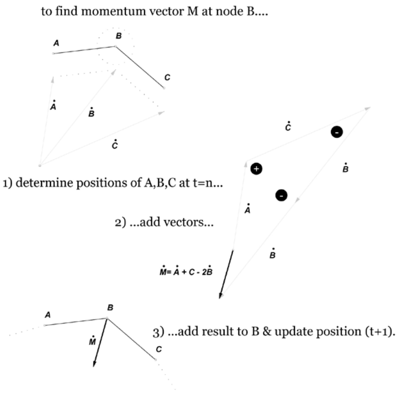
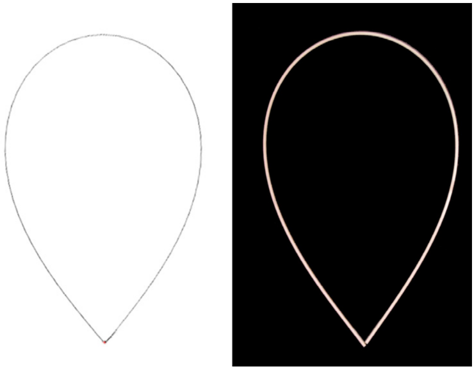

to find momentum vector M at node B....
A
B
C
$\dot{A}$
$\dot{B}$
$\dot{C}$
1) determine positions of A,B,C at t=n...
$\dot{C}$
-
$\dot{B}$
+
-
$\dot{A}$
2) ...add vectors...
$\dot{B}$
$\dot{M}= \dot{A} + C - 2\dot{B}$
A
B
$\dot{M}$
C
3) ...add result to B & update position (t+1).

Figure 7
Qualitative comparison between simulation result (left) and physical model (ABS, 2mm circular cross-section, L=700mm).



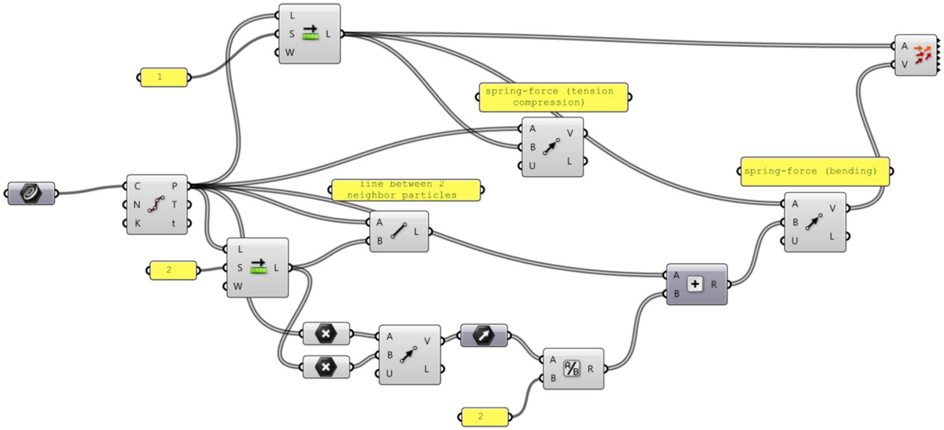

1
spring-force (tension compression)
line between 2 neighbor particles
spring-force (bending)
2
2
L
S
W
L
C
N
K
P
T
t
A
B
U
V
L
A
V
A
B
R

Figure 8
Visualization scheme for analysis of momentum-vectors. (Needs arbitrary curve as input).

It is not the intended goal of a physics-based modelling design tool for architectural application to capture any material property perfectly and simulate it in real-time. Rather the existing engineering solutions have shown that the amount of input parameters necessary to run precise simulations is a hindrance for some designers in early stages. The success of the hanging chain model and its virtual counterpart, the spring-based particle system, is its simplicity. As it is a simulation of generic behaviour (things fall down / hang down) rather than specific material properties, it becomes a design tool (Figure 7). The previously described method is a simple expansion of the successful, yet radically simplified

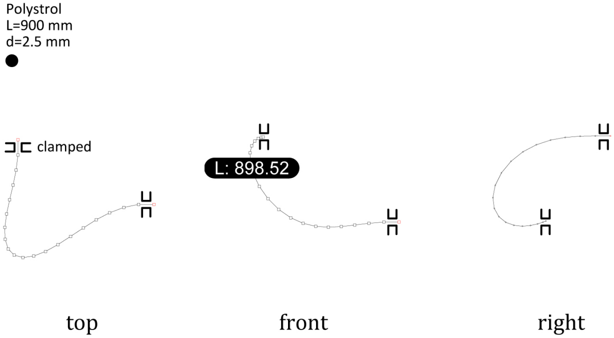


Figure 9
Spatially deformed elastica, constrained at endpoints (Top, Front, Left view).

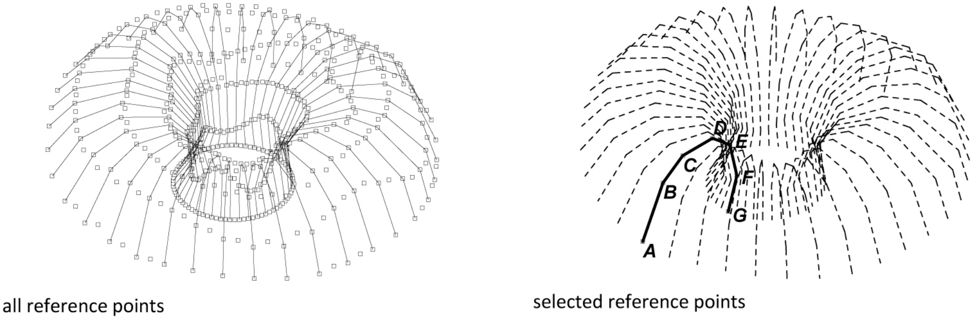


Figure 10
Geometric polyline model and reference nodes for comparison (left) and selected polyline (right).

spring-based particle system based on Hooke's law.

The "stiffness" of a material is represented by an additional vector of a particle-based spring system. In a chain of springs, this vector acts on each particle and tries to align it with its two neighbors.

Similar to the four generic Euler cases we now have a design framework, for linear elements that buckle under compressive load. We achieve this without the simulation of any specific material property, but through approximating the formal shape a buckling element WANTS to take. Through piecewise discretization into sets of three nodes, this mechanism works in R2 and R3 (Figure 7 and 9). It might be, that in the real word, due to cross-sectional dimensions and material choice the formal design goal of such a simulated element cannot be achieved, but this is not a limitation of the design tool. Rather the design space is more focused, by implementing a generic material behaviour in early design stages through embedded formal behaviour. This behaviour is generic and therefore true for all stiff/material (similar to the Euler case). Through further refinements during a design process, the designer can customize the tool and limit the simulated shapes to match the material characteristics of a chosen material and Crossection.

Advantages

The advantages of the above described method of expanding a spring-based particle system by an additional vector that is able to simulate bending resistance for linear elements are first of all that described mid-point method is very simple to understand - even from a designer's perspective. Furthermore it is also because it is vector-based, relatively easy to visualize in tools such as processing. This does not mean it could not be realized as a purely numerical calculation. This flexibility is another important advantage in terms of how the method could be adopted and embedded into larger (more material specific) simulation frameworks.

Another important aspect is that we have shown that a regular spring-based particle system can be expanded by an addition force simply by adding another vector to each particle and updating its position. While this aspect per se is not surprising, as it formulates the basis of how a particle system works, we would like to highlight the flexibility. This means we can start to add even more vectors, e.g. for self-weight, once we have determined more specific materials and geometries of our simulated system. As a sandbox for the development of a particle-system processing has proven to be very versatile as it not only provides vector-classes combined with graphical output in an accessible object-oriented programming language. It also features an integrated update function via the void draw method that lends itself for the simulation of dynamic systems. This is a great advantage over arguably more accessible node-based graphical user interfaces such as Grasshopper for Rhino, or Generative Components, that are built to run once and update based on user interaction. Physics-engines in the form of Kangaroo and some other Plugins that

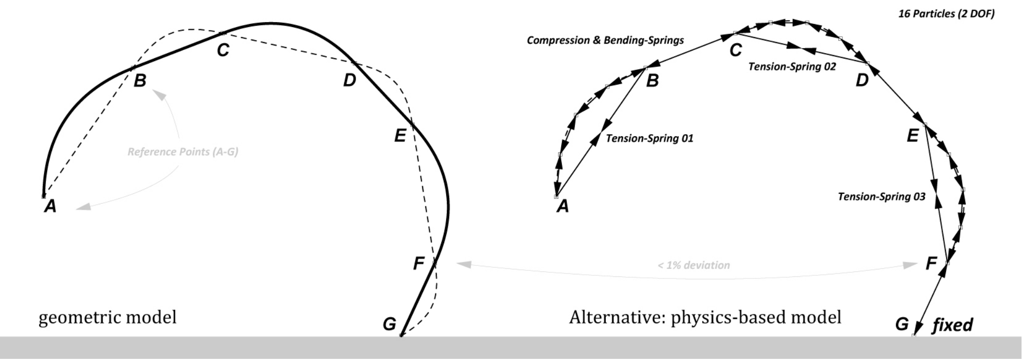


Figure 11
Comparison between reference node positions in geometric model (left) and physics-based simulation model (right).

are built on top of parametric design packages such as Grasshopper benefit from their accessibility, yet lack the ability to start "sequenced" simulations (do this, THEN do that), something that becomes quite crucial, when looking at construction sequences of force-active structures.

Disadvantages

As with any Particle-System, there is the question how the simulated data translates into the real-world. As forces are determined based on node positions, the simulated shapes have actually much more in common with the diagrams in force equilibrium known from Graphics statics that with the colorful visualization of van-mies stresses that are often seen from analysis carried out in FEM-Simulations. Because of the lack of "real" material properties and dimensions, the room for interpretation of the simulation is more difficult and leaves more room for speculation. The designer's task is to use this generic simulation of systemic behaviour and carry out further development in the software, to make it suit his needs. This task is individually different and cannot be the same for every simulated system.

The challenge is to understand something that behaves like a physical object on the screen, isn't linked to a material framework such stiffness, mass or self-weight. At least in the current state, where for example the "weight value" for a node is usually set to "1" for computational costs. Also, while some might argue, that a particle system that is accessible via source-code in processing is a flexible and extendable base, most designers still find it inaccessible.

CASE STUDY

The ICD/ITKE Pavilion 2010 was a bending-active shell structure. It was built with the use a script, that generated the entire geometry based upon a carefully controlled polyline model (Figure 10). The geometry of the polylines determined the overall geometry of the pavilion and was calibrated through extensive material tests. We wanted to test an alternative computer-based model for determining the pavilion's geometry in real-time based upon the physics-based modeling approach, described prior.

Comparing the locations of the reference nodes, we found that the above method is able to simulate the geometry of a representative cross-section and could very well serve as an alternative modeling strategy in terms of precision compared to the geometric model (Figure 11). Yet the effort to set up such a physics-based model and the time it takes to calculate it might render it unfeasible for such simple structures such as the pavilion. Also, some precautions have to be mad to ensure, that all the elements buckle into the correct directions (same as with physical experiments). Because of the low curvature of the bent elements in the original pavilion, it was not only valid, but actually very effective to use a b-spline interpolation for these regions and constrain the geometric variations to the limits determined

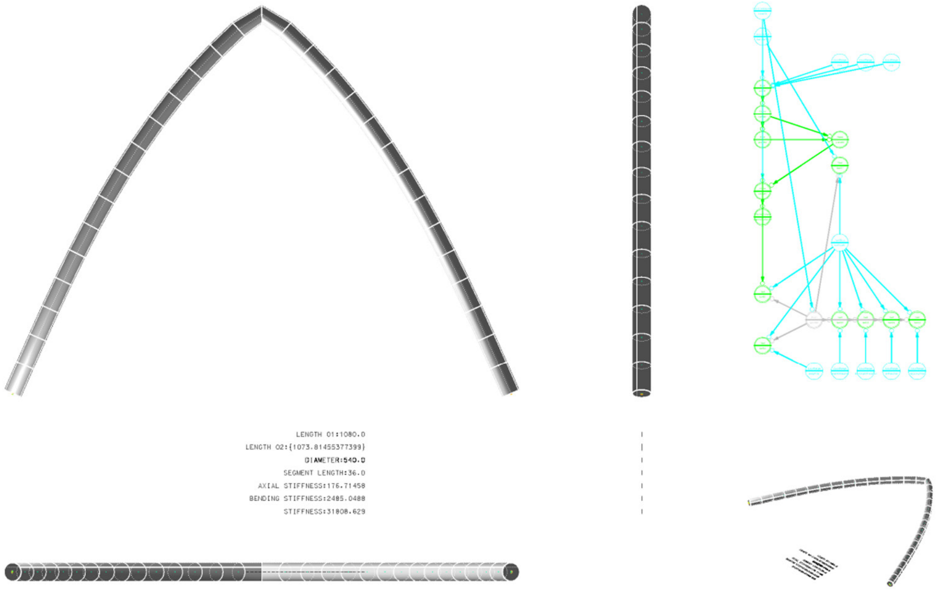

LENGTH 01:1080.0
LENGTH 02:{1073.81455377399}
DIAMETER:540.0
SEGMENT LENGTH:36.0
AXIAL STIFFNESS:176.71458
BENDING STIFFNESS:2485.0488
STIFFNESS:31808.629

Figure 12
"Kink" in elastica model: by linking the simulation to geometric information such as crossectional values and variations thereof, more articulated geometries can be produced directly within the parametric modelling environment.

through physical material tests. Other, more complex bending-active structures, such as gridshells with 2-dimensionally deformed elastic lattices and spatially deforming elements, might benefit more from this alternative modeling approach. The same can be said for more from such an approach, yet the above described benefits are applicable to this model as well and we are carrying out tests, to compare whether the abstract physics-based model is able to represent the actual physical geometry more closely (deformations under self-weight, openings,...) than the idealized geometric model did.

CONCLUSION

Numerical methods that depend on the understanding of mathematical syntax and methods such as integration and matrices are difficult to read for a designer with a common architectural education. Vector-based methods for force simulation are more accessible by someone, who is used to produce visual output such as drawings, diagrams and sketches. As part of particle systems, these vector-based methods have been successfully applied as equivalents of physical hanging-chain models and for the simulation of tension-active structures such as cablenets and to a certain extend - membranes.

By incorporating additional vectors to account for bending-resistance, self-weight and other forces, the simulations can be further developed from merely behavioral systemic simulations toward specific material- and geometry-driven real-time force diagrams that help the designer to develop force-active material systems and connect to far more advanced simulation environments that are necessary for the development of architectural systems on larger scales.

OUTLOOK

The above described physics-based modeling strategy with linear elastic springs including springs for the simulation of bending resistance should not be measured against scientific simulations - rather these simulations resemble real-time force diagrams similar to the reciprocal diagrams. In these diagrams, developed as part of the scientific field of graphics statics, forces are drawn as vectors - exactly as done here. In graphic statics, the requirement for the calculations to be correct is that the sum of all vectors of a system need to equal 0. This leads to a closed funicular polygon in the reciprocal diagram of a static system.

Here the system is dynamic, which means that we allow the positions of the nodes to change in order to (almost) converge to force equilibrium. As seen in other dynamic relaxation processes, the speed at which the process converges depends on a variety of parameters and can vary largely depending on the skill and knowledge of the designer.

Despite this immanent parallel to the scientific calculations carried out graphically in graphic statics, here, in the computer-based processing sketch, we do not read force vectors as real external forces, which are directly linked to factors such as self-weight, which are determined by material choice and geometry. This obvious extension can be next step of developing a more "architectural" physics based simulation - also for bending active systems.

Another approach could be to not only balance external and internal forces by solving for a force equilibrium via vector-addition, but to look at the more material-specific internal stresses, this produces. Here an approach could be to introduce material-specific non-linear springs which's stress-strain curve is directly linked to a specific material characteristic.

Also a more direct interaction between a parametric design software and the simulation tool (similar to Kangaroo for Grasshopper) could be realized- this would allow to link geometric information to the simulation parameters and vice versa (Figure 12)..

The simulation of bending in the described framework is limited. The twisting that occurs by bending an element in R3 is not accounted for. It could be developed in another step, or we introduce interpolated b-splines (that have the information about twist embedded as part of their frenet frame) through the particles that serve as a feedback.

While we have shown a simulation of a linear element, the framework allows to expand the simulation to 2-dimensional spring networks investigate the behavior of plate-like elements.

REFERENCES

Adriaenssens, S 2001, "Tensegrity spline beam and grid shell structures," *Engineering Structures*, 23(1), pp. 29-36.

Baraff, D and Witkin, A 1998, "Large steps in cloth simulation," *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, pp. 43-54.

Block, P and Ochsendorf, J 2007, "Thrust Network Analysis: A New Methodology For Three-Dimensional Equilibrium", *Journal of the International Association for Shell and Spatial Structures (IASS)*, 48(3), pp. 167-173.

Eberly, DH 2004, *Game physics*, Morgan Kaufmann Publishers, Heidelberg.

Fröhlich, P 1995, *FEM-Leitfaden*, Springer, Berlin.

Gaß, S 1990, Experimente - Form <-> Kraft <-> Masse 5, *Physikalische Analogmodelle im architektonischen Entwerfen*, Krämer, Stuttgart.

Gibson, SFF and Mirtich, B 1997, *"A survey of deformable modeling in computer Graphics"*, Tech. rep., MERL - A Mitsubishi Electric Research Laboratory.

Kilian, A and Ochsendorf, J 2005, "Particle-Spring Systems For Structural Form Finding," *Journal Of The International Association For Shell And Spatial Structures (IASS)*, 46(147), p. 77.

Knippers, J, Cremers, J, Gabler, M and Lienhard, J (eds) 2011, *Plastics and Membranes Construction Manual*, Birkhaeuser Architecture, Basel.

Lachauer, L, Jungjohann, H and Kotnik, T 2011, "Interactive Parametric Tools for Structural Design," *Proceedings of the IABSE-IASS Symposium*, pp. 126-133.

Levien, R 2009, "*From Spiral to Spline: Optimal Techniques in*

Interactive Curve Design", PHD, Engineering–Electrical Engineering and Computer Sciences, University of California, Berkeley.

Li, J and Knippers, J 2011, "Form-finding of grid shells with continuous elastic rods", *Proceedings of the IABSE-IASS Symposium*.

Otto, F and Schleyer, F-K 1962, *Zugbeanspruchte Konstruktionen - Band II*, Ullstein GmbH, Frankfurt / Berlin.

Otto, F, Schauer, E, Hennicke, J, and Hasegawa, T 1974, *Gitterschalen*, Karl Krämer, Stuttgart.

Ramtin, A and Aish, R et al 2010, "Embedded Rationality. A Unified Simulation Framework for Interactive Form-Finding," *International Journal of Architectural Computing (IJAC)*, 8(3), pp. 399-418.

Walser, A 2011, "*Formfindung von Schalen mit numerischen Hängemodellen*", Institut für Baustatik und Baudynamik, Universität Stuttgart.

[1] http://web.media.mit.edu/~amanda/gaudi/

[2] http://people.bath.ac.uk/abscjkw/ComputerPrograms/ProcessingPrograms/Bending/

[3] http://spacesymmetrystructure.wordpress.com/

[4] http://www.food4rhino.com/project/kangaroo

[5] www.grasshopper3d.com/